实验编号： 4 **四川师大《IOS》实验报告 2018** 年 **9** 月 **26** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 自定制视图、文件、Playground可视化

姓名：\_蒋星\_ 学号：2016110418 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验\_四\_ \_**自定制视图、文件、Playground可视化**\_**

1. 实验目的及要求
2. 掌握沙盒文件的操作；
3. 掌握自定制视图的定义及使用；
4. 掌握playground下可视化开发方法；
5. 实验要求
6. 认真填写实验报告，要求附加部分运行界面和主要代码；
7. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
8. 实验内容
9. **文件缓存处理**
   1. 判断沙盒的Document目录下是否存在某文件夹，如果没有则新建一个该文件夹；
   2. 判断是否该文件夹下存在一个图片文件，如果存在该文件，读取该文件到一个图片对象中并进行显示，如果不存在则从网上下载一张图片并保存到该图片文件中；
10. **自定制视图**
    1. 从UIView中派生一个自定制的View；
    2. 绘制一个椭圆（或则自己喜欢的任何图形）；
    3. 新建视图对象并进行显示；
11. **代码版hello world（在视图控制器中加入代码）**
    1. 代码中生成label（outlet）和button；
    2. 将label和button加入根view中；
    3. button添加像self（target）发射action（selector）的操作；

实现clicked响应代码（selector）

1. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
2. **文件缓存处理**
   1. 判断沙盒的Document目录下是否存在某文件夹，如果没有则新建一个该文件夹；
   2. 判断是否该文件夹下存在一个图片文件，如果存在该文件，读取该文件到一个图片对象中并进行显示，如果不存在则从网上下载一张图片并保存到该图片文件中；

* 程序代码：

// documents 下面是否存在此文件夹

func DocumentsIsHaveDirectory(\_ Name:String)->Bool{

    let isExist\_path = NSHomeDirectory() + "/Documents/\(Name)";

    return  FileManager.default.fileExists(atPath: isExist\_path)

}

print("是否存在文件 www: \(DocumentsIsHaveDirectory("www"))")

//创建文件夹 函数

func CreateDirectoryAtDocuments(\_ Name:String){

    let path = NSHomeDirectory() + "/Documents/\(Name)";

    let manager = FileManager.default

    if(!manager.fileExists(atPath: path)){

        try! FileManager.default.createDirectory(atPath:path, withIntermediateDirectories: true, attributes: nil);

    }

}

CreateDirectoryAtDocuments("www")

print("是否存在文件 www: \(DocumentsIsHaveDirectory("www"))" )

enum PictureType{

    case JPG

    case PNG

}

func DownloadPictureFromInternalSaveInPath(\_ pictureURL:String, \_ pictureType:PictureType,\_ store\_path:String) -> Bool{

    let imageUrl = URL(string: pictureURL)

    let imageData = try! Data(contentsOf: imageUrl!)

    let imageUI = UIImage(data: imageData)

    let picture\_store\_Path = NSHomeDirectory() + store\_path;

    var data:Data;

    switch pictureType {

        case .JPG:

            data = UIImageJPEGRepresentation(imageUI!,1)!

        case .PNG:

            data = UIImagePNGRepresentation(imageUI!)!

    }

    try? data.write(to: URL(fileURLWithPath:picture\_store\_Path))

    return UIImage(contentsOfFile: picture\_store\_Path) != nil; //检测是否存储成功

}

let isOk =  DownloadPictureFromInternalSaveInPath("http://t2.hddhhn.com/uploads/tu/201707/521/85.jpg", PictureType.JPG, "/Documents/www/my.jpg")

if isOk{

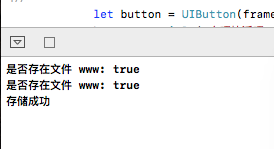
     print("存储成功")

} else {

    print("存储失败")

}

* 运行结果：



1. **自定制视图**
   1. 从UIView中派生一个自定制的View；
   2. 绘制一个椭圆（或则自己喜欢的任何图形）；
   3. 新建视图对象并进行显示；

* 程序代码：

//: Playground - noun: a place where people can play

import UIKit

import PlaygroundSupport

var str = "Hello, playground"

class MyView:UIView{

    override func draw(\_ rect: CGRect) {

        let path = UIBezierPath(ovalIn:rect)

        UIColor.red.setStroke()

        path.stroke()

        UIColor.gray.setFill()

        path.fill();

    }

}

class TriangleView:UIView{

    override func draw(\_ rect: CGRect) {

        let path = UIBezierPath();

        path.move(to: CGPoint(x:rect.size.width/2,y:0));

        path.addLine(to: CGPoint(x:rect.size.width,y:rect.size.height))

        path.addLine(to: CGPoint(x:0,y:rect.size.height))

        UIColor.green.setFill()

        path.fill()

    }

}

let vw = MyView();

vw.draw(CGRect(x:0,y:0,width:200,height:100))

class Controller:UIViewController{

    var lable:UILabel!

    var circleView:MyView!

    var tangleView:TriangleView!

    override func loadView() {

        view = UIView(frame:CGRect(x:0,y:0,width:400,height:800));

        view.backgroundColor = UIColor.white;

        lable = UILabel(frame:CGRect(x:50,y:100,width:300,height:50))

        lable.backgroundColor = UIColor.blue

        lable.textColor = UIColor.white

        lable.textAlignment  = .center

        lable.text = "hello the laji yu yan swift"

        view.addSubview(lable)

        circleView = MyView(frame:CGRect(x:100,y:400,width:200,height:200))

        circleView.backgroundColor = UIColor.green

        view.addSubview(circleView)

        tangleView = TriangleView(frame:CGRect(x:100,y:180,width:200,height:200))

        tangleView.backgroundColor = UIColor.red

        view.addSubview(tangleView)

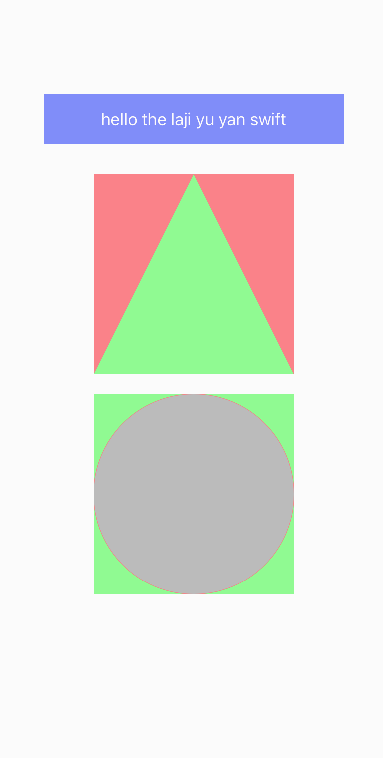
    }

}

let ct = Controller()

PlaygroundPage.current.liveView = ct

* 运行结果：



1. **代码版hello world（在视图控制器中加入代码）**
   1. 代码中生成label（outlet）和button；
   2. 将label和button加入根view中；
   3. button添加像self（target）发射action（selector）的操作；

实现clicked响应代码（selector）

* 程序代码：

class MyController:UIViewController{

    var lable:UILabel!

    @IBAction func clicked(){

        print("i am clicked");

        lable.text = "Hello World";

    }

    override func loadView() {

        view = UIView(frame:CGRect(x:0,y:0,width:400,height:700));

        view.backgroundColor = UIColor.red;

        lable = UILabel(frame:CGRect(x:50,y:100,width:300,height:50));

        lable.backgroundColor = UIColor.blue;

        lable.textColor = UIColor.white;

        lable.textAlignment  = .center;

        lable.text = "点击发现我要说的话";

        view.addSubview(lable);

        let button = UIButton(frame:CGRect(x:140,y:500,width:120,height:40))

        button.setTitle("来呀快活呀", for: .normal);

        button.addTarget(self, action: #selector(MyController.clicked), for: UIControlEvents.touchUpInside);

        button.backgroundColor = UIColor.white;

        button.titleLabel?.textColor = UIColor.black;

        view.addSubview(button)

    }

}

let ct\_3 = MyController()

PlaygroundPage.current.liveView = ct\_3

* 运行结果：



1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

Github地址：https://github.com/KikcerGoer/study-swift/tree/master/homework

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。